Exhibit 24

Free and Open Source Java

Free Range.

Check the terrain for new opportunities.

Get the Source »

FAQ

The Announcement

Q: What is Sun announcing?

A: Sun is open sourcing all of its Java platform implementations under same license (called GPL Version 2) used by the GNU/Linux operating system.

Specifically Sun is announcing:

- GPL v2 license for Sun's Java SE (JDK) and Java ME implementations, and adding this license to Sun's Java EE implementation.
- First release of code for the JDK and for Sun's Java ME implementation, projects and communities.
- Roadmap for future code releases and community development.

This singular act is the largest contribution ever made to the free software community, and places Sun squarely at the front of the open-source movement - as the single biggest commercial contributor.

Back to top

Q: What is the significance of this announcement to the industry?

A: This announcement celebrates the open sourcing of the code base for one of the industry's most significant and pervasive software platforms, to foster adoption in new markets, to build broader communities, and fuel even more innovation. With over 3.8 billion Java technology enabled devices, Java technology has already demonstrated explosive growth, appearing in volume nearly everywhere. Now, as free software, the Java platform can address new markets and be the engine of innovation for the next generation of networked applications.

Back to top

Q: Why did you choose the GPL for Java? Why not the CDDL that Sun created for OpenSolaris? Why not the Apache Software License?

A: One of our primary goals in this move was to grow the Java market, benefiting everyone - platforms, ISVs, OEMs, customers of every type as the reach and richness of the Java market grows. We considered a number of different license choices, including CDDL. While each of the open-source licenses we considered had a positive potential impact, our evaluation brought us to the conclusion that using the GPL would result in the greatest incremental growth to the Java market.

Back to top

Java SE (JDK) Announcement Specifics

Q: What is the OpenJDK Community?

A: Sun is establishing the OpenJDK Community for the ongoing development of Sun's open-source implementation of Java SE. The OpenJDK Community is where developers will gather to collaborate on the open-source JDK code base and related projects. The OpenJDK project in which the open-source code base will ÒliveÓ is part of this community. Through the OpenJDK project, developers will be able to directly influence the future of the JDK implementation, participate with their peers in an open community and help take Java technology where it hasn't been before. Sun is evolving the existing JDK Community, where developers have been working on the source code over the past two years, into a site where Sun and non-Sun developers alike can collaborate together on the implementation. The OpenJDK Community can be found at http://community.java.net/openjdk.

Back to top

Q: What components of the JDK software are you open sourcing today?

A: We're open sourcing the Java programming language compiler ("javac"), and the Java HotSpot virtual machine. In addition, we're open sourcing the JavaHelp 2.0 extensible help system, Sun's reference implementation of JSR 97 in this first code release. The JavaHelp code base is found at https://javahelp.dev.java.net.

Back to top

Q: Is the JDK source code under the GPL as well?

A: Yes, the JDK is licensed under the GPL version 2 with the Classpath exception. This license, which will be used for all of Sun's Java implementations, is endorsed by the Free Software Foundation, and is the license used by the GNU/Linux operating system.

Back to top

Q: When will you finish open sourcing the JDK? What is the timeline?

A: We expect to release a fully buildable JDK based almost completely on open-sourced code in the first half of 2007.

Back to top

Q: Will Sun continue distributing its commercial implementation of the JDK?

A: Yes. For people who want the benefits of commercial support, and predictability, they may choose to use Sun's commercial distribution of the JDK or JRE. The free commercial implementation of the JDK may be found at http://java.sun.com/javase/downloads/index.jsp. Similarly, Sun's free JRE may be downloaded from http://java.com. To learn more about development and deployment support options, visit http://sun.com/javasupport.

Back to top

Q: Will Sun's commercial JDK releases be built from the open-source code?

A: Yes, for the most part. Since there's some encumbered code in the JDK, Sun will continue to use that code in commercial releases until it's replaced by fully-functional open-source alternatives.

Back to top

Q: What components of the JDK software are you open sourcing today? Why did you choose these components?

A: We're open sourcing the Java programming language compiler ("javac"), and the Java HotSpot virtual machine. In addition, we're open sourcing the JavaHelp 2.0 extensible help system, Sun's reference implementation of JSR 97 in this first code release. The JavaHelp code base is found at https://javahelp.dev.java.net. Sun will release all of the JDK as noted in the next few quarters as we work through encumbrance issues.

Fortunately, we have these major subcomponents of the JDK that are both unencumbered and that embody some of the most significant innovations in Java technology.

Back to top

Q: Are you open sourcing the Java language or the Java SE platform specifications?

A: We are not open sourcing the Java programming language, nor the platform APIs and specifications, which are governed by the JCP. We're open sourcing Sun's implementations of the Java SE and Java ME specifications.

Back to top

Q: Which version of the JDK do these components come from? Are you open sourcing the latest code?

A: We're open-sourcing these components from a very early build of JDK 7. In order to prepare these components to be open sourced, we not only changed the license text but we also simplified the build process in order to make these components more easily buildable outside of the full JDK source tree. JDK 6, is nearly finished, hence we're releasing these components from the JDK 7 tree. The only other differences between the JDK 6 and 7 versions of these components are minor bug fixes and enhancements that have already been integrated into the JDK 7 tree. When we open-source the full JDK we'll make the sources for both JDK 6 and JDK 7 available. The community will have both a stable release - JDK 6 - on which to focus quality improvements, and JDK 7, the next feature release where all the action will be for innovation and new capabilities.

Back to top

Q: What is the Java compiler ("javac")? What can developers do with this code?

A: javac is the compiler that translates Java programs into "byte codes" that are then executed by a Java virtual machine (JVM). Developers will be able to experiment with the compiler, fix bugs, and try out new optimizations with this early code drop.

Back to top

Q: What is the Java HotSpot virtual machine? What can developers do with this component?

A: Java HotSpot is Sun's implementation of the Java virtual machine. The Java HotSpot VM includes the core execution engine for the Java platform, including:

- dynamic compilers that convert Java bytecodes into optimized native machine code on supported hardware platforms
- memory management and garbage collection subsystems
- threads and synchronization
- monitoring, debugging, and profiling telemetry
- parts of the Java security architecture including the bytecode verifier

As such it is the component of the Java SE platform that delivers most directly on the "Write Once, Run Anywhere" promise of Java technology.

Developers will be able to learn how a world-class virtual machine is built, fix bugs, experiment with new garbage collection, synchronization, and bytecode compiler algorithms, and port the VM to new hardware architectures and operating systems with this code.

Back to top

Q: What is JavaHelp? What is the relationship of JavaHelp to the JDK?

A: JavaHelp software is a full-featured, platform-independent extensible help system that lets you incorporate online help in applets, components, applications, operating systems, and devices. Authors can also use JavaHelp software to deliver online documentation for the Web, and for corporate intranets. JavaHelp is Sun's reference implementation for JSR 97.

While JavaHelp is not part of the JDK, it is a closely related technology, and is being open sourced at the same time as the first code components of the JDK.

Back to top

Q: Where do developers go to participate in implementing these JDK components?

A: Developers can gain immediate access to these components under the open-source license by visiting https://openjdk.dev.java.net.

Back to top

Q: How can I get involved?

A: Visit the OpenJDK Community at http://community.java.net/openjdk and check out the projects, mailing lists, pointers to blogs, and other opportunities to participate. Sun, and indeed the whole Java technology ecosystem, welcomes any and all suggestions, bug fixes, and contributions of code, ideas, and energy. Once the full JDK is available in the OpenJDK project, there will be more opportunities, but even now there are plenty of ways to roll up your sleeves and dig in to the code base.

If you'd like to check out the entire source code for JDK 6 and JDK 7 under the Java Research License now, while waiting for Sun to complete the process of open sourcing the JDK, you are welcome to visit and join these projects at https://jdk6.dev.java.net and https://jdk7.dev.java.net respectively.

Back to top

Q: What is the governance model for the OpenJDK projects?

A: Initially the governance will be similar to what has been in place for the existing JDK Community. Sun will be evolving the governance model to one of community involvement, with a goal of inclusive, meritocratic governance.

Back to top

Q: How does the governance and infrastructure of these first open-source JDK projects differ from the existing JDK Community and jdk6 and jdk7 projects on java.net?

A: We've made several improvements to the contribution process to simplify the process for getting developers' bug fixes and implementation ideas into the code base.

Back to top

Java ME Announcement Specifics

Q: What is the Mobile & Embedded Community?

A: The Mobile & Embedded community establishes a central location for the open-source development of Java ME technologies and applications. This community has been launched on java.net and can be found at the following URL: http://community.java.net/mobileandembedded. It currently includes the phoneME project, the cqME project, and the application developer project. In the future we anticipate that this community will grow by adding additional Java technology and application projects.

Back to top

Q: What license did you choose for Java ME?

A: Java ME code will be licensed under GPL v2, as are all of our Java implementations.

Back to top

Q: What is Sun open-sourcing in Java ME?

A: Sun is open sourcing its implementations of Java ME. Immediately available at launch will be the source code for Sun's feature phone implementation called the Sun Java Wireless Client (based on the Connected Limited Device Configuration, CLDC), the next generation version of the platform that currently enables rich mobile data services in over 1.5 billion handsets. Later this year, Sun will release its advanced OS phone implementation (based on the Connected Device Configuration (CDC) specification).

Sun is also open sourcing its compatibility and quality testing tool frameworks. This includes the source code for the Java ME TCK Framework, the foundation for Sun's Java ME compatibility tests. We believe this can standardize the industry on a single framework to simplify the testing process.

In coming months, Sun will open-source the Java Device Test Framework, the foundation for the quality and function tests. This project will enable developers to create new tests, reducing implementation variation and enabling their applications to run across multiple devices.

Back to top

Q: What is the feature phone implementation?

A: The feature phone implementation is a Java runtime designed to run on today's mass-market handsets.

Back to top

Q: What is the Advanced OS phone implementation?

A: The Advanced OS phone implementation is a Java runtime designed to run on operating systems targeting advanced mobile devices like smartphones, set-top boxes, etc.

Back to top

Q: Will Sun's implementation be built from open-source code?

A: Yes, Sun's implementations for feature phones and advanced OS phones will be based on the open-source code base.

Back to top

Q: What can a developer do immediately with the CDC and CLDC code bases?

A: A developer will be able to immediately build both the feature phone and the advanced OS phone code bases once these are open sourced. Developers will have the opportunity to download, evaluate, and play with the source code, and help in its ongoing development.

Back to top

Q: Will Sun continue to ship commercial implementations?

A: Yes, Sun will continue to ship commercial implementations for feature phones and advanced OS handsets under Sun's commercial licenses.

Back to top

Q: What is the difference between the open-source and the commercial code bases?

A: The differences include encumbrances and some minor modifications to the source code such as splash screen, logos, license, header files, etc.

Back to top

Q: What can a developer do with the framework code bases?

A: A developer could use these frameworks to drive testing of compatibility tests for new mobile JSRs, and create tests to improve the quality of implementations.

Back to top

Q: Where do developers go to participate in the Java ME projects?

A: The Mobile & Embedded community has been created to enable platform developers and application developers to participate. Visit http://community.java.net/mobileandembedded for more information.

Back to top

Q: What is the phoneME project?

A: The phoneME project is where Sun is releasing its phone implementations for Java ME. It has a single repository that consists of various active development modules including CLDC, CDC, MIDP, and various JSR implementations.

Back to top

Q: What is the cqME project?

A: The cqME project (compatibility and quality) is where Sun has released the source code and is doing the active development on the Java ME TCK Framework. In the future, Sun will release into this project the source code and engage in the active development of the Java Device Test Framework.

Back to top

Q: What is the Application Developer project?

A: The Application Developer project will provide resources to developers and a place to engage in the development of open-source Java ME applications. The project will be the home for the new developer guidelines that were created in partnership with Orange. These guidelines will help developers minimize porting efforts. This project will also be home for open-source Java ME applications.

Back to top

Java EE and NetBeans Announcement Specifics

Q: How does this announcement affect Java EE?

A: Sun's implementation of Java EE 5 has been available as open-source under the CDDL license through the GlassFish Community since June of 2005. In order to gain all of the benefits of the GPL v2 license and to be able to offer implementations of the entire set of Java platforms under the same license, the GlassFish application server source code will be made available under the GPL v2 license with Classpath exception in addition to the CDDL, By adding a second license, we simplify the process of combining and distributing GlassFish code with other GPL licensed communities. By offering all of Java under a common license, developers can now more easily collaborate on and distribute updated versions of Java SE, Java EE, and Java ME together. For more information about GlassFish, go to http://java.sun.com/javaee/community/glassfish/index.jsp.

Back to top

Q: How can developers use the NetBeans IDE to get started in working with this code?

A: The javac sources have already been configured as NetBeans projects, so all you have to do is download them, open them in the NetBeans IDE, and use the Build Project command to build them. For further information and a step-by-step tutorial go to http://nb-openjdk.netbeans.org/netbeans.org.

In addition, the NetBeans Mobility Pack makes it easy to create mobile applications with drag-and-drop screen design. The world-record producing Sun Studio development environment lets you work on the platform-specific native code in the Java HotSpot virtual machine.

Back to top

General and Goals

- What influenced this decision?
- Why is this good for Java developers?
- How does this benefit Java customers and users?
- What markets do you believe this initiative will open up for the JDK?
- What impact will this move have on the adoption of the Java platform?
- What are your goals in open sourcing Sun's Java ME implementations?
- Who benefits from open sourcing Java ME?
- What does Sun mean by "compatibility?"
- Do you think anyone will fork the JDK?
- So, what about compatibility? How will Java technology remain "Write Once, Run Anywhere" after Sun's Java platform implementations are open sourced?

Q: Why is Sun open sourcing its Java implementations now?

A: The Java platform is 11+ years in the making. When it was first released it was a radical departure for commercial software, including full source code under a novel license. Sun and the Java ecosystem have been extremely successful at establishing and growing a very large and dynamic market in this time. The Java platform retained its own licensing model even as the open-source model proliferated, because the Java licensing model successfully created a large and open market with many compatible choices. We now see an opportunity to encourage even more possibilities for adoption in places the Java platform hasn't gone before, where Free licensing and open-source development approaches are a prerequisite for consideration.

At the same time, the Free software and open-source communities are now saying that compatibility is a given for any Java implementation. And there is a new spirit of innovation with Web 2.0, SOA and collaboration/participation technologies, and the Java platform is the perfect foundation platform on which to innovate - and open-source can help accelerate this innovation.

Back to top

Q: What influenced this decision?

A: Key Free software and open-source communities have stated that they believe that only Java technology implementation that are completely compatible with the specification can succeed in the market. These communities, which provide thought leadership for the open-source Java world, are now focused on delivering only compatible implementations.

In addition, Sun has gained experience in community development with the Project GlassFish open-source implementation of Java EE, with OpenSolaris, with OpenOffice.org and NetBeans, and with the JDK Community on java.net. This experience makes us confident that when we open-source Sun's Java implementations, the platform will benefit, and we can better balance the needs of community with those of customers, end users, and licensees.

Overall, we feel that caution is appropriate for a technology that affects the lives and livelihoods of tens of millions of people around the world. After much reflection we feel now is the perfect time to take the next step with the Java platform.

Back to top

Q: Why is this good for Java developers?

A: Because volume wins. Open sourcing Sun's Java SE implementation will lower barriers to adoption in markets where open-source software leads. As new markets adopt Java technology, developers will discover new opportunities. More applications. Innovations that leverage the industrial strength foundation of Java to deliver valuable new products and services. And developers will be able to directly influence the future of the JDK implementation, participating with their peers in an open community. Taking Java where it hasn't been before, and helping to ensure that Java technology remains a central unifying standard for the Internet.

Back to top

Q: How does this benefit Java customers and users?

A: Your investment is safe, with multiple independent implementations of Java SE, and the reference implementation from Sun available under an open-source license. An open-source JDK provides peace of mind, plain and simple:

- You can adopt Sun's Java technologies with full confidence. Java SE will be freely available - subject to the market forces that drive competition, reduce price and speed innovation.
- Whether you value Free and open-source software for philosophical or economic reasons, want more flexibility, appreciate the quality that transparency brings, or want to know you have ultimate control over your investment, you can adopt the JDK knowing that you'll gain these advantages.

- With the rock-solid, high performance JDK implementation from Sun under the same open-source license used by GNU/Linux distributions, you can expect Java technology to find its way onto new platforms, be leveraged for new applications and infrastructure, and be the foundation of tomorrow's most exciting new products and web technologies.

Your investment in Java technology will become even more valuable as open-source speeds innovation, spreads adoption, and carries the platform to places it couldn't previously go.

Back to top

Q: What markets do you believe this initiative will open up for the JDK?

A: Several markets are likely to find the JDK to be more suitable for use once it is under an open-source license:

- Government, educational, and research markets where open-source software is either mandated, or highly desired.
- Enterprises that mandate open-source infrastructure to retain maximum flexibility and drive competitive procurement.
- Enterprises and organizations that have selected OS distributions based on GNU/Linux and OpenSolaris as preferred OSs for deployment.

In order to gain the benefits of commercial support, and predictability, these customers may choose to use Sun's commercial distribution of the JDK or JRE in conjunction with a support contract. However, they would not consider the commercial product if the open-source version were not available.

Back to top

Q: What impact will this move have on the adoption of the Java platform?

A: The Java platform has been very widely adopted already - it is one of the most important and widely used components of modern web-based infrastructure. But there remain un-tapped or under-served markets. In particular, Java technology is not always included in open-source web infrastructure ÒstacksÓ that are commonly distributed and deployed alongside and included with GNU/Linux distributions. Those that do include Java technology often do not include an up-to-date, compatible runtime and development environment. We will be working with the GNU/Linux distributions to get the JDK included as part of the free software repositories commonly included with these open-source operating system distros. Once the JDK is easily obtained and installed with these platforms, we expect to see more widespread adoption of Java technology outside of North America and Western Europe, as well as in cutting edge web infrastructure deployments worldwide.

Since the GPL is a very widely used open-source license (in fact it's the same license used by GNU/Linux), distributing the JDK under the GPL with GNU/Linux distributions should be a good match, making it easier to adopt by those looking for open-source alternatives.

Back to top

Q: What are your goals in open sourcing Sun's Java ME implementations?

A: To remain the number one mobile application development platform*, Java ME needs to grow and evolve. This means engaged developers, accelerated innovation, and a more consistent implementation across devices. [* Evans Data Corp. Spring 2006]

Back to top

Q: Who benefits from open sourcing Java ME?

A: Everyone in the Java ME ecosystem benefits from open sourcing Java ME. Benefits include:

- simplified evaluations,
- transparent development,
- opportunities to directly influence the future of the platform

By open sourcing these implementations, handset manufacturers can leverage the common code base to reduce their development costs. The result will be less variation across devices Đ in other words, ÒWrite Once, Run AnywhereÓ taken to the next level.

Both application developers and operators will benefit from the accelerated innovation, enabling developers to continue to create compelling applications and services, which in turn help drive revenue. Operators and handset manufacturers will benefit from lower porting, testing and maintenance costs.

Back to top

Q: What does Sun mean by "compatibility?"

A: Compatibility for a Java technology means the implementation of that technology meets the

associated compatibility requirements of its Technology Compatibility Kit or TCK. For Java SE this means passing the TCK tests and other requirements defined in the Java SE TCK.

Back to top

Q: Do you think anyone will fork the JDK?

A: We expect great new ideas and valuable research to come from forks to the platform. In addition, Sun encourages compatible forks where the code is ported to additional hardware and software platforms. Such ports extend the breadth of Java SE to places currently not supported by any vendor. Broad distribution of incompatible forks is potentially a danger since such forks could damage the "Write Once, Run Anywhere" compatibility value of the Java platform.

Back to top

Q: So, what about compatibility? How will Java technology remain "Write Once, Run Anywhere" after Sun's Java platform implementations are open sourced?

A: The Java technology compatibility promise is so central to the value of the platform that it has been the single most important influence in driving the detailed planning and decisions for this initiative. Sun is making a number of key decisions and commitments to the community that we believe will help foster compatibility:

- License: GPL makes proprietary forks less likely, as all changes must be published.
- Branding: the Java trademark and logos are for compatible implementations only.
- JCP: the JCP's role as the governing body for Java standards and evolution is not changed by this move.
- Community Development: Sun's experience in building strong communities means developers will likely find the governance and infrastructure to their liking.

In addition to the specific steps that we're taking to help foster compatibility, we are convinced that the market will demand compatible implementations, and that incompatible ones won't gain traction. With the billions of dollars of Java applications in the installed base, the community has recognized that a Java implementation that doesn't run the installed base of code won't get very far. That is why both the GNU/Classpath and Apache Harmony projects have stated unequivocally that they're working to build fully compatible implementations of Java SE.

We have experience in the Java EE world to back this up: there are four compatible, open-source Java EE implementations in the market including Sun's GlassFish Application Server, and no incompatible variants have achieved any market penetration. We expect the same dynamic to happen with both Java SE and Java ME platforms.

Back to top

Business Model

Q: What do Java SE customers buy from Sun?

A: The JRE and JDK binaries themselves will remain zero cost downloads. End-user customers will buy subscriptions and one-time engagements for enterprise-class support from Sun for Java SE, including:

- High quality, optimized performance binaries released on a predictable schedule.
- Packaged updates, patches, and security fixes.
- Break/fix support including customer-specific bug fixes delivered out of cycle.
- Branding: Access to trademark and logo licenses including "Java Powered".
- Installation, custom engineering services.
- Training, education.
- Developer support.

Commercial source code licensees may optionally buy additional services and gain additional rights, including:

- The right to ship binaries derived from custom modified source code.
- The right to make modifications to the source code without putting those modifications back into the open-source code commons.
- The right to use the "Java Compatible" logo for compliant custom implementations.
- Porting services.
- Custom development and engineering support services.
- Upgrades and updates to the source.
- Performance tuning services.

- Binary support.
- Testing services, including TCK testing.
- Training.

In addition, some customers needing specialized versions of Java SE for embedded and real-time applications may purchase licenses to these added-value implementations. We're also investigating support services that could be offered optionally to customers who prefer to Òdo it themselvesÓ using the open-source code base, but need a bit of additional help.

Back to top

Q: Why purchase these added value services and implementations from Sun?

A: Sun, the only systems vendor to have open sourced nearly its entire software portfolio, enables customers to leverage a number of unique advantages that only it can offer, including:

- The economic advantages of the open-source business model through a larger range of the stack.
- Recognized Ògold standardÓ Java SE and Java ME implementations on which the open-source communities are eager to innovate.
- Choice of GPL license that maximizes the value of the open-source code commons and maximizes incentives for compatibility, while creating an opportunity to monetize commercial licenses and support services for licensees.
- The ability to tap the key architects and creators of the JDK, including for support and tuning services.
- Systems optimized to run Java applications with superior throughput, power, and space characteristics.
- Development tools including NetBeans 5.5 and Sun Studio that deliver increased productivity for developers.

In addition, Sun brings unique expertise and knowledge of both Java technology and open-source best practices to the table. This combination will help Sun establish OpenJDK as the most compelling open-source community for innovation, allowing Sun to gain maximum benefit from the open-source initiative. Some of these advantages include:

- Experience at both understanding and balancing the needs of the diverse and complex Java ecosystem.
- Unmatched developer outreach and community development expertise.
- World-class ability to combine enterprise-class rigorous development processes and open-source transparency, bringing together the best of community development with top-quality, on-time delivery of commercial products.
- Proven success at delivering scalable community infrastructure that manages the largest open-source projects on earth.
- Vision and commitment to participation, transparency, open development, and collaboration.
- Deep understanding of F/OSS community concerns and aspirations, and a practical perspective on balancing the needs of community with the interests of customers.
- Best practice policies and procedures for taking large code bases open source, balancing concerns and monetizing adoption and ubiquity, all while respecting intellectual property rights.

Back to top

Q: How will Sun's Java ME business be affected?

A: Sun is confident that licensees will continue to leverage Java ME commercial products and services that help them deliver an exciting and compelling mobile internet experience to consumers. As a result, we expect our business opportunities to multiply based on the improved economics that easy access to source code will bring.

Back to top

License

Q: What license did you choose for the open-source JDK components?

A: GPL v2 for almost all of the virtual machine, and GPL v2 + the Classpath exception for the class libraries and those parts of the virtual machine that expose public APIs.

Back to top

Q: What license did you choose for implementations of Java ME and the Java ME frameworks?

A: GPL v2 was chosen for all components related to Java ME.

Back to top

Q: Did you use the GPL v2 verbatim, or did you alter it?

A: No alterations. Sun is using the unmodified GPL v2.

Back to top

Q: Is GPL v2 an open-source license?

A: Yes, see http://opensource.org/licenses/gpl-license.php.

Back to top

Q: What is the Classpath exception?

A: The Classpath exception was developed by the Free Software Foundation's GNU/Classpath Project (see http://www.gnu.org/software/classpath/license.html). It allows you to link an application available under any license to a library that is part of software licensed under GPL v2, without that application being subject to the GPL's requirement to be itself offered to the public under the GPL.

Back to top

Q: Why do you need the Classpath exception?

A: If an application is distributed with an implementation of Java such as the JDK under GPL v2,

that application could be subject to the requirements of the GPL that all code that is shipped as part of a Òwork based on the [GPL] programÓ also be GPL licensed. Accordingly, a GPL license exception is needed that specifically excludes from this licensing requirement any application that links to the GPL implementation. The Classpath exception accomplishes this. Without the Classpath exception, a Java SE implementation licensed under GPL v2 could not practically be distributed with non-GPL licensed Java applications. This could present a serious barrier to adoption, for example by OpenSolaris or GNU/Linux distributions if left unaddressed.

Back to top

Q: Why did you choose this licensing method?

A: This is the licensing paradigm in common use within Free software communities such as GNU/Classpath and Kaffe for the components of a Java technology implementation including the virtual machine and class libraries. We consciously chose the same licensing method so that there would be no temptation to second guess Sun's intention to make its Java SE implementation available under a genuinely Free and open license.

Back to top

Q: Doesn't GPL v2 + Classpath exception offer very similar licensing terms to the LGPL? Why not use LGPL instead?

A: Yes, from a practical perspective the Classpath exception establishes similar terms to LGPL. However, the Free software Java technology communities haven't chosen to use LGPL, and so we at Sun decided to follow their lead and use the Classpath exception.

Back to top

Q: Does this mean Sun is abandoning CDDL?

A: Not at all! Sun has developed a broad and pragmatic approach to Free and open-source licensing, and uses several different licenses to meet the varying needs of the communities it has started.

For example, note that GlassFish adds (does not replace) GPL v2 with the Classpath exception to the CDDL license under which it now is offered.

Back to top

Q: Will you license the entire JDK under this method when you open-source the rest of the code base?

A: Yes.

Back to top

Q: How can you ship the JDK with binary-only elements then? You said there were encumbrances.

A: Well spotted! None of the Java SE components included in this initial announcement are encumbered, so there are no binary-only elements. If, at the time we make the full JDK or other Java SE components available, there may still be encumbered components that must be shipped without source. The Software Freedom Law Center and the Free Software Foundation have helped us craft a special exception to the GPL to allow the full JDK to be built. This exception will be applied temporarily until the encumbrances are removed. We would welcome your help to make this happen as soon as possible.

Back to top

Q: Why didn't you choose a license like BSD or Apache v2?

A: Sun had several objectives in mind in choosing the license for the JDK source code. We wanted to:

- Minimize the likelihood of incompatible forks.
- Drive more adoption.
- Engage a broad cross-section of the open-source communities.
- Protect and enhance the investments of those who have licensed and chosen to support the Java platform.

After extensive analysis and consultation with experts both inside and outside of Sun, we determined that of the choices available, the GPL v2 + Classpath exception license paradigm is most likely to achieve these objectives. We know that this choice is unlikely to please everyone but we believe it offers the best balance of opportunity for developers and Sun.

Back to top

Q: Why is the license different for Java ME than for the JDK? Why no Classpath exception for Java ME?

A: With Java ME, there is no way to install and integrate implementations. Instead, implementations are integrated with the hardware. As such, the problem the Classpath exception is solving with the Java SE implementation isn't present with Java ME.

Back to top

Q: What are the advantages of GPL v2?

A: As well as fulfilling its original purpose of promoting Free software, the GPL v2 is designed to help advance projects and code commons by requiring innovation sharing with the commons. By design, it minimizes proprietary forks by requiring any modifications be shared with the project. GPL v2 is the right license to preserve Java's trademark "Write Once, Run Anywhere" value proposition.

Back to top

Q: What rights do developers have under GPL v2 + Classpath exception?

A: The best source for answering this question is the license itself (http://www.gnu.org/licenses/gpl.html). In addition, there have been numerous analyses of the GPL license and its terms, not least by the license stewards, the Free Software Foundation (http://www.fsf.org/). Sun recommends studying these interpretations, and consulting legal counsel as necessary to understand your rights under this license.

Back to top

Q: What are developers' obligations under this license?

A: Once again - the best source for understanding Sun's chosen license for the JDK source code is the license itself.

Back to top

Q: How can someone use the open-source code base to create a derivative work?

A: Sun's Java platform implementations are now Free software. Developers can use the open-source JDK and Java ME code bases in any way that a GPL v2 licensed code base may be used. These implementations are not "special cases".

Back to top

Q: Can someone create an implementation that isn't compatible with the Java specification using this code?

A: Yes. We do not recommend or endorse that action, however. In addition, they cannot label that implementation with the Java Compatible or Java Powered brand and logo. These brands are your assurance that an implementation has passed the relevant TCKs.

Back to top

Q: Can someone create software that doesn't even implement Java, but uses pieces of the JDK? What are the limitations, if any?

A: Yes. There are no limitations. But there is an obligation, as with any code licensed under GPL v2, to ensure that all resulting programs that incorporate the pieces used are accompanied by full source code as required by the license.

Back to top

Q: Can I call the resulting software "Java"?

A: No.

Back to top

Q: What must I do to call my software based on code from the OpenJDK or phoneME projects "Java"?

A: The requirements for the use of the "Java" trademark and name have not changed with the open sourcing of the JDK and Java ME source code. The GPL v2 does not include a trademark license - no OSI-approved open-source licenses do. Sun does not currently have a licensing program that permits the use of the "Java" mark in your product or company name. You can use a truthful "tagline" however associating your product or company with Java technology, according to Sun's standard terms for use for trademarks. Please see http://www.sun.com/policies/trademarks/ for more details.

Back to top

Q: What must I do to use the "cup and steam" logo? Does Sun license it?

A: Sun offers several logo programs featuring the distinctive Java "cup and steam" logo design. To see whether you qualify for one of these programs and to learn how to participate, please visit http://java.com/brand for more details.

Back to top

Q: How do I know which components in the OpenJDK project are under just the GPL, and which are under the GPL + Classpath exception?

A: Each source file is individually licensed. You can see the exact file headers we're using here: http://openjdk/space/Operations+Team/License+Details. The key difference is the presence of the sentence:

> Sun designates this particular file as subject to the "Classpath" exception as provided by Sun in the LICENSE file that accompanied this code.

in the "+ Classpath" version. So basically if you see the word "Classpath" in the license header then you know that the exception applies.

Back to top

Q: What about GPL v3? Have you considered using that license?

A: While Sun has been working with the Free Software Foundation as an active participant in the development and review of the GPL v3 license, this license is not yet complete. It is Sun's strong desire to complete the open sourcing of its Java technology implementations in a timely manner, so we made the decision to use an existing, established license paradigm rather than wait for GPL v3 to be completed. Using GPL v2 does not indicate anything negative about GPL v3. Sun continues to be very actively and positively involved in this new license's development.

Back to top

Q: I want to contribute. Do I need to sign anything to get started?

A: Yes. Sun requires that contributors to all of its Free and open-source projects sign the Sun Contributor Agreement (SCA) and mail or fax back the completed agreement. A copy of the SCA can be found at https://jdk.dev.java.net/Sun_Contributor_Agreement_v1.2.pdf.

Back to top

Q: Why do you have a Contribution Agreement?

A: The Sun Contributor Agreement (SCA) is needed to ensure that Sun has the rights to use your contributions in products and projects. It also asks you to warrant that you are entitled to grant Sun these rights, and that, to your knowledge, your contribution does not violate anyone else's rights. Sun's responsibilities to existing licensees are not removed by Freeing these implementations, so Sun needs this sharing in order to serve the extended Java community.

Back to top

Q: Do I lose any rights to my contribution under the SCA?

A: Unlike some other contribution agreements that require you to transfer copyrights to an organization, the SCA does not take away any of your rights to your contributed intellectual property. When you agree to the SCA, you grant Sun joint ownership in copyright, and a patent license for your contributions. You retain all rights, title, and interest in your contributions and may use them for any purpose you wish.

Back to top

Q: But other communities are non-profits. Why should I share my copyrights with Sun?

A: Sharing copyrights protects Java Community interests. Sun acts as a bridge, ensuring that both open-source communities and traditional commercial organizations are able to grow the Java ecosystem, for the benefit of all. To do this, Sun needs to be able to offer the Java platform as a whole under both open-source and commercial licenses. The SCA enables Sun to do that. Additionally it allows Sun to ensure that the origins of every line of code are known, in case of future litigation against Sun or the community, and it allows Sun to upgrade the open-source licenses in the future should that become necessary.

Back to top

Q: What can Sun do with my contribution?

A: Sun may exercise all rights that a copyright holder has, as well as the rights you grant in the SCA to use any patents you have in your contributions. So may you.

Back to top

Q: Will Sun continue to offer the JDK and Java ME source code under a commercial license,

along with GPL v2?

A: Yes. Indeed, some of Sun's existing licensees will continue to prefer a commercial license over an open-source license for a variety of reasons.

Back to top

Q: Why will Sun continue to offer the JDK and Java ME sources under a commercial license?

A: Sun has commercial obligations towards its licensees which of course we will continue to satisfy after the code base is open sourced. Sun will continue to work with licensees who want to distribute derivatives of Sun's implementations without the requirement in the GPL to contribute modifications back to the community. Sun's commercial Java technology distribution licenses require licensees to distribute only compatible implementations as determined by the TCK for relevant Java technologies.

Back to top

Q: What is the process to get a commercial license for the JDK, Java ME implementations, or Java ME TCK Framework source code?

A: For more information on obtaining a commercial license to the JDK source code, please contact Sun Sales at http://www.sun.com/sales/wwsales.jsp.

Back to top

Q: Will Sun continue to offer its binary bundles for the JDK and JRE?

A: Yes, Sun will continue to offer these bundles under the Binary Code License (BCL). In addition, Sun will offer Distribution License for Java (DLJ) bundles for GNU/Linux and OpenSolaris distributions.

Back to top

Q: Is Sun changing the terms of the Binary Code License for Sun's JRE?

A: Not at this time.

Back to top

Q: What about the JCP's Spec License? Is that changing?

A: No, the Spec License is not changing.

Back to top

Q: Is Sun changing the terms under which it licenses the Java SE and Java ME TCKs?

A: Not at this time. Our efforts have been focused on making the source code available.

Back to top

Q: Can I combine a virtual machine derived from the open-source Java HotSpot code and an existing Sun Java class library?

A: The release candidate builds of JDK 6 contain special language in their distribution license (section 3.6) which specifically allows using the Sun Java class library with open-source Java HotSpot for internal and evaluation purposes only (note: the release candidate software is not authorized for production use nor for redistribution) This is a temporary work-around to allow effective evaluation and testing of the open-source Java HotSpot code. Once the class libraries have been open sourced, this work-around won't be needed.

Back to top

Q: How can Sun have other licenses that bear on the open-source code base? Isn't that no longer "open source"?

A: Because Sun owns the copyright for the open-source code base, Sun is able to license each copy of this code base distributed by Sun, under any license, including a commercial software license. This right is inherent in copyright law. Several Free and open-source communities exhibit this behavior.

Back to top

Q: Will Sun offer binary builds of the open-source JDK code base for download? If so, under what license?

A: The initial modules being released in the first code release for the JDK will not have binary builds available. We're examining this question for the major code release in Spring, 2007, and will be considering encumbrances when deciding.

Back to top

Code and Encumbrances

Q: What do you mean by "encumbered code"?

A: Some parts of the JDK and Java ME implementations include code to which Sun may not hold sufficient rights to release under a license that allows third parties to create unlimited sublicensable derivative works. These rights are necessary to allow release of these implementations under the GPL v2. Those parts of the code for which we don't have these rights are said to be "encumbered".

Back to top

Q: What do you mean when you say that you'll ship a "fully buildable" JDK in early 2007?

A: We'll release as much of the source code as possible under the GPL. It seems inevitable however that there will be encumbered modules in the first buildable release of the JDK. In the short term we'll provide a separate bundle of binaries compiled from these sources that can be combined with builds of the GPL'd sources to yield a complete, working JDK. In the long term we hope to rewrite these components to remove the closed-source code - and we welcome your help!

Back to top

Q: What are the encumbered components in the JDK?

A: The most significant encumbrances are in the 2D graphics area: The ICC color-management library, the font rasterizer, and the graphics rasterizer. There is a fairly direct open-source replacement for the color-management library, so we suspect that one will be easy. Reasonably good open-source font and graphics rasterizers are available, but unfortunately they don't at present support all of the features of the Java 2D API, so fitting such components into the JDK will require significant work.

Back to top

Q: How will the encumbered binaries be licensed? Will I be able to redistribute a JDK built from the GPL'd sources and the encumbered binaries?

A: We will be using a special GPL v2 exception written with assistance from the Free Software Foundation and Software Freedom Law Center that will allow downstream redistribution of such a JDK. For more on this subject see "Licensing".

Back to top

Q: Will everything in the Java ME feature phone and advanced OS phone implementations be available in open source?

A: No, there may be several encumbrances in these phone implementations. These encumbrances include support for specific phone hardware, graphics engine, sound engine, etc. All of these components and any other encumbered code will not be included in the open source code base.

Back to top

Q: How will encumbered code in the Java ME code base be managed?

A: Sun is actively negotiating with Intellectual Property (IP) owners to gain the rights to add the encumbered code, if any, to the open source code base. We will try to clear encumbrances with open-source or other reasonable alternatives. Implementations including encumbered components will continue to be made available via commercial license.

Back to top

Governance

- Will Sun always be in control of what goes into the JDK?
- Will there ever be non-Sun committers to the JDK?
- Will non-Sun committers to the JDK be treated as equals?
- Will it be possible to contribute code to the OpenJDK community that Sun does not plan to ship in its implementations, such as ports to other operating systems or hardware platforms?
- Does Sun plan to make its JDK quality test suites available?
- Will Sun allow alternative implementations of components to exist in the OpenJDK code base?
- What's the Mobile & Embedded community governance model?
- Can you create a new project within the Mobile & Embedded community?
- Can I contribute fixes and enhancements to the phoneME and cqME projects?
- Who will decide which contributions are accepted to the Mobile & Embedded Community projects?
- Will there ever be non-Sun committers to the phoneME and cqME projects?
- Will non-Sun committers be treated as equals?
- Does Sun plan to make its Java ME quality test suites available?
- What is the criteria for becoming a full Project in the Mobile & Embedded community?

Q: What's the community governance model for the OpenJDK Community?

A: We're studying existing governance models in the open-source community, but we want to take the time to do a thorough analysis and gather more input and advice before making this very important decision. Our plan is to evolve the governance model to one of community involvement, with a goal of inclusive, transparent, meritocratic governance.

Back to top

Q: Can I contribute fixes and enhancements to the JDK between now and then? If so, how?

A: Certainly! Based on our experience with the transparent development initiative for JDK 6, we've devised a new, lighter-weight process for handling contributions. Please visit https://openjdk.dev.java.net/contribute.html for more details.

Back to top

Q: Who will decide which contributions are accepted to the OpenJDK code base?

A: As with the JDK 6 initiative, the final decision will rest with Sun during this interim period.

Back to top

Q: Will Sun always be in control of what goes into the JDK?

A: We are committed to building a thriving community around this code base so that the proven power of open development can be brought to bear on the JDK. Yet we're also determined to retain the ability to ship high-quality releases in a timely fashion, a goal that's of benefit not just to Sun but to the entire Java community. We are confident that these two goals can be reconciled, and look forward to the community's advice on how to respect both of these objectives in making decisions on code changes.

Back to top

Q: Will there ever be non-Sun committers to the JDK?

A: Yes! We are very excited about collaborating with non-Sun community members on the code, and as is customary with open-source projects, inviting the best of them to become committers to the code base on equal terms.

Back to top

Q: Will non-Sun committers to the JDK be treated as equals?

A: Yes. It is our expectation that they'll have all the rights that Sun engineers have. They'll also have all of the responsibilities, including code-review duties and the obligation to adhere to, as well as to assist, the various development practices and processes that help to build a high-quality JDK.

Back to top

Q: Will it be possible to contribute code to the OpenJDK community that Sun does not plan to ship in its implementations, such as ports to other operating systems or hardware platforms?

A: Probably, but this is an area where we could use community input. Sun can't test such code, so it might not have the same level of quality as the rest of the code base. This is further complicated when unrelated changes are made that inadvertently cause problems in such contributed code.

Back to top

Q: Does Sun plan to make its JDK quality test suites available?

A: We'll ship some of our test suites, but not all, and not all at once. As much of the regression and unit-test suite as is practical will be made available along with the code. We plan to make some of our functional tests available eventually. Some tests have dependencies on internal test frameworks, test harnesses, or environment which are not available outside of Sun, which may limit their use by the community. We also plan to create a quality portal in order to work closely with the community on testing.

Back to top

Q: Will Sun allow alternative implementations of components to exist in the OpenJDK code base?

A: Probably. But as with code that will not be part of Sun's implementations, synchronizing unrelated changes could be difficult. We could use some input to help us determine how to manage such things.

Back to top

Q: What's the Mobile & Embedded community governance model?

A: The Mobile & Embedded community will launch with an interim governance model that will evolve over time. The Sun team has studied many of the existing governance models in the open-source community and has determined what it believes is a good initial model to follow.

The model is based on its founding principles of transparency, participation, compatibility and engineering excellence. It consists of a governing board of 5 members. 2 are permanent seats for representatives from Sun, 1 is appointed by Sun (non-Sun member), and 2 are elected from the membership of the Mobile & Embedded community. The role of the board is to maintain the health of the community by overseeing its affairs and facilitating the alignment between the operations of the community and its established principles and objectives.

The Mobile & Embedded community will be made up of Java Platform Projects (based on JSRs), Java Tools projects (tools, utilities, libraries, etc) and Java application projects (midlets or xlets). New projects can be established in the community by filling out the Project Request forms and meeting the entry criteria. Approved projects are placed in the incubator. A project remains in incubator status until it meets the criteria for becoming an official Mobile & Community project. Project level governance is determined by the project owner.

Back to top

Q: Can you create a new project within the Mobile & Embedded community?

A: Yes, new projects can be created in the Mobile & Embedded community. New projects get created by submitting a new project proposal on java.net. These newly requested projects that meet the incubator entry criteria get placed in the incubator. To graduate from the incubator to official Mobile & Embedded Community project status, a project must meet certain criteria such as transparency, quality, compatibility requirements, etc.

Back to top

Q: Can I contribute fixes and enhancements to the phoneME and cqME projects?

A: Certainly! We strongly encourage contributions into these new projects. In order to be able to contribute the following steps needed to be completed:

- Sign and return the Sun Contributor Agreement (SCA)
- Apply for a community login on java.net
- Become familiar with the community and projects.
- Submit your contribution.

If you are a committer, submit your contribution through Issue Tracker. If you are a contributor, find a committer to work with you through the submission process. More information on how to contribute can be found at the following URL: https://mobileandembedded.dev.java.net/content/contribute.html.

Back to top

Q: Who will decide which contributions are accepted to the Mobile & Embedded Community projects?

A: Committers have the authority of submitting code into the source tree. Until an individual becomes a committer, contributors will have to work with a committer in order to get their code reviewed and submitted.

Back to top

Q: Will there ever be non-Sun committers to the phoneME and cqME projects?

A: Yes! As with the OpenJDK Community, we very much want to collaborate with non-Sun community members on the code. We would hope to make the most skilled participants committers to the code base.

Back to top

Q: Will non-Sun committers be treated as equals?

A: Yes. As with the OpenJDK project, non-Sun committers will have all the rights and responsibilities that Sun engineers have.

Back to top

Q: Does Sun plan to make its Java ME quality test suites available?

A: Sun has no plans to make its quality test suites available.

Back to top

Q: What is the criteria for becoming a full Project in the Mobile & Embedded community?

A: The criteria for becoming a full Project in the Mobile & Embedded community is alignment with the Founding Principles and meeting the necessary legal requirements.

Back to top

The Java Brand

Q: What is the Java brand?

A: The Java brand has two components: its name, Java?, and its logo, the Java cup and steam. In addition, there are "sub-brands" that further clarify your use of Java technology and define how a product complies with its brand promise: Java Powered? and Java Compatible?.

These brands - and their logos - are valued registered trademarks owned, protected, and managed by Sun Microsystems. The Java brand carries with it a very specific value proposition that tech-savvy consumers and IT professionals everywhere recognize - a promise of "Write Once, Run Anywhere", and an expectation of exciting, robust, easy to use applications generated from its code.

Sun Microsystems will continue to own and protect the brand and its value by managing its use. Any product that wishes to use one of the family of Java brands must comply with the associated Java brand program. These are outlined at http://www.java.com/en/about/brand/.

Back to top

Q: Can I call products I create from code I download from your open-source site "Java?"?

A: If your product meets one of several program requirements for using the Java brand, including the applicable testing, (see http://www.java.com/en/about/brand/) then you can use the appropriate Java cup and steam logo, according to the guidelines of the brand.

For example:

- "My product, Foo Sneakers is Java Compatible." (add logo)
- "This derivative of the OpenJDK project source code is Java Compatible." (logo)
- "I created my application using a Java Compatible implementation and it is Java Powered" (logo)

The Java Powered logo requires that you be a member of the Sun Partner Advantage Program and ship a Java-based application for Java SE 1.4 or later. The Java Compatible logo program for Java SE requires that your implementation passes the relevant TCKs for Java SE, and that you apply for the logo. If however, you choose not to pursue and meet the requirements of a Java brand program, then you are not granted rights to use the Java name or its logo. Instead, your use of the word Java is limited only to what is legally referred to as "fair use." (See question below). Please note that under US law, and other jurisdictions, there is no "fair use" of a logo, and you must have a license.

For example, you can say:

- My product uses code I downloaded from the openjdk.dev.java.net website.
- My product, JCool for Java Technology, can be used with Java SE 6.

Back to top

Q: Can I use the term "JDK" to describe my product - it's built with code from the OpenJDK Project after all?

A: The trademark "JDK" is Sun's name for the Java Development Kit. Accordingly, it follows similar rules to the "Java" trademark. You may not refer to your product or implementation as a JDK, even if it is based on code you downloaded from the OpenJDK Project. You can use the term "JDK" under Sun's "fair use" guidelines, however. Please refer to Sun's trademark policy for more information.

Back to top

Q: How do I know if someone else's product is compatible?

A: If a product carries the Java cup and steam logo, and/or uses the term Java Compatible?or Java Powered?, then it is bound by Sun's trademark and compatibility program rules, and carries the guarantee of a tested, compatible product.

If a product describes a relationship to an open-source Java community and its code, such as "derived from code found at openjdk.dev.java.net", then it is important you look further into its claims, and ask for more concrete evidence that the code has been tested and certified as "Java powered or "Java Compatible." For instance, you might ask if the solution has been certified by the appropriate TCK test suite.

Back to top

Q: How can I describe a product I create using code from the open-source site if I don't wish to get it certified by Sun?

A: If you create a derivative product from code you download from one of the open-source Java technology project sites on java.net such as openjdk.dev.java.net, it is not considered "Java Compatible" until it is certified; it is instead, a "derivative work based on the open-source code from the OpenJDK Project." While it may truly be a derivative of Java technology, trademark law only supports use of the word if it's verifiable and licensed by Sun.

Trademark law supports however, a notion of "fair use" that allows the use of a trademarked word as text under certain guidelines.

When making fair use of a Java trademark, you should acknowledge that Sun Microsystems owns the trademark. The following language is appropriate:

"Java, JDK, OpenJDK, Java Compatible, Java Powered, and ... are trademarks or registered trademarks of Sun Microsystems, Inc. in the U.S. or other countries."

For a comprehensive description of how and when you are allowed to use the Java trademark, please refer to http://www.sun.com/policies/trademarks/.

Back to top

Q: What is the Java Verified Program?

A: The Java Verified Program is a wireless industry driven organization that provides a complete testing process for mobile Java technology applications. The member companies consisting of Motorola, Nokia, Siemens, Sony Ericsson, Vodafone Group, LG Electronics, Orange and Sun Microsystems create the testing criteria by which the mobile applications are tested. Those applications that pass the certification testing gain the right to use the Java Powered Logo and a digital certificate to prove the authenticity and integrity of the software.

Back to top

TCKs

Q: What are TCKs?

A: Technology Compatibility Kits (TCKs) are the compliance tests, tools and documentation which allows you to establish whether a particular implementation completely and correctly

implements a particular Java technology specification as defined by a Java Specification Request (JSR) on a specific host platform. If your product or implementation passes the TCK for a specification, and meets all the untested compatibility requirements for a specification described in its TCK, it compatibly implements that specification.

Back to top

Q: Will developers have the ability to run TCK tests on components they modify?

A: We know that access to the TCK tests is important to developers working on the open-source code base. However, we've been completely focused on the mechanics of getting the initial source code release completed, and at this time do not have any information to provide on TCK access.

For some time Sun has offered the Compatibility Testing Scholarship Program as a way for qualified educational institutions and non-profits to gain access to TCK's and support services. The TCK Scholarship program will continue.

Back to top

Q: What is the procedure for an individual or organization to apply for and receive a TCK support scholarship from Sun, through the JCP?

A: Please see the section "How to apply for a scholarship" on the website.

Back to top

JCP

Q: Will open-sourcing Java SE and Java ME require any changes in the JCP?

A: No. Many open-source implementations of JSRs developed under the JCP are available today.

For example, Sun's reference implementation of JSR 244, Java EE 5, has already been open sourced as the GlassFish application server.

Nonetheless, the JCP itself is constantly reviewing its own rules to ensure they are fair and modern and indeed, is currently engaged in revisions - see below.

Back to top

Q: Where will Java SE and Java ME specification evolution occur?

A: The JCP defines the process whereby Java technology specifications are created and updated. This will not change. Today, Sun is the Spec Lead for the umbrella JSRs which define the components of Java SE. The Java SE Expert Group works with the Spec Lead to determine which additional JSRs and other minor improvements will be included in each new release of the specification. Final approval for all changes is made by a vote of the JCP Executive Committee. The same process holds true for the Java ME platform.

Back to top

Q: How can I influence the evolution of Java SE or Java ME?

A: There are several ways to influence the direction of Java SE or Java ME platforms: 1) you can join the JCP and propose or join a JSR which will be included in Java SE or Java ME specifications, 2) comment on such JSR specs during public reviews, 3) work with Sun or a member of a JSR Expert Group to sponsor your ideas, 4) work with Sun in the OpenJDK or Mobile&Embedded communities to develop minor enhancements to the implementation (i.e., small improvements not warranting a separate JSR). The JCP Program Office has recently sponsored JSR 306 which in part will look into ways to get more individual involvement in JSRs.

Back to top

Q: How do I join the JCP?

A: Join the JCP by signing the Java Specification Participation Agreement (JSPA). You can find the JSPA and instructions on how fill it out and send it in here:

http://jcp.org/en/participation/membership.

Back to top

Q: My employer won't allow me to join the JCP as an individual. I still would like to participate in Java's evolution, so what can I do?

A: Many employers have policies about employee involvement with outside groups such as the JCP or other standards bodies. There are often good reasons for such policies and your employer should be able to explain them to you. As described above, however, there are ways to participate without signing the JSPA. In addition, one of the goals for JSR 306 is to look for means to increase participation of individuals and to allow more outside influence into the JSR development process.

Still, as a non-member you have many options to participate and influence: all spec drafts are publicly accessible, and you can provide direct feedback at any time to the spec lead and expert group by emailing the comments alias you can find on each JSR's web page.

Last but not least, in order to stay informed about the Java Community Process, you may subscribe to the JCP mailing list: JCP-INTEREST. Visit the web site for more details.

This list will allow you to follow the progress of specification proposals as they go through the Community Process. You will receive messages when new specifications are proposed, when revisions to existing specifications are requested, when specification drafts are available, and when the specifications are finalized.

Back to top

Community Development and Infrastructure

Q: What's required to participate in the OpenJDK and Mobile&Embedded Communities?

A: Nothing but your time and interest! You can visit openjdk.dev.java.net or mobileandembedded.dev.java.net and look around, join mailing-list discussions, download the binaries and file bugs, or take the source to make improvements and/or use it yourself. It's up to you. You don't even need a java.net user ID to join the mailing lists. Please let us know if you have any suggestions.

Back to top

Q: Do I need to sign something before making a contribution?

A: Yes. If you want to contribute code for inclusion in the OpenJDK, phoneME, cqME projects, or other Sun-sponsored open-source projects, then we ask that you sign the Sun Contributor Agreement (SCA) first so that Sun has the right to distribute your contribution if accepted. This is a standard practice in many open-source communities. If you've already signed an SCA for another Sun project, this agreement covers your participation in any of the projects discussed above.

Back to top

Q: Do I give up any rights by signing that agreement?

A: No, you merely commit to sharing your rights in the contributed code with Sun. For more information about the Sun Contribution Agreement please check under the "Licensing" section.

Back to top

Q: What will Sun be doing to attract developers to participate in its JDK open source effort?

A: Initially Sun engineers will continue to interact with external developers through mailing lists, blogs, and 1:1 e-mail conversations as we do now. The compiler and Java Hotspot VM projects on openjdk will have their source refreshed periodically to mirror the same source

base used for development within Sun. Sun engineers associated with those components will also work with developers who show interest. Over the next year or so Sun will make more of its internal processes, mailing lists, and materials available to the developer community.

Back to top

Q: What's the holdup on delivering tools and processes for the OpenJDK community? What can't they be made available now?

A: Most of the tools and processes we use are Sun-specific. For example, our source control management and bug tracking systems are proprietary and not usable outside Sun's infrastructure. We're going to change that, but it will take some time. Our processes also need to be documented in a manner that can be understood by people who don't work at Sun. Our top priority is to get the source code out and the rest will come as quickly as possible afterwards.

Back to top

Q: What tools will the Mobile & Embedded community use for collaborative development?

A: The Java ME team is utilizing all of the collaborative tools offered on java.net including the source control management system, bug tracking, blogs, forums, mailing lists, etc to enable community development of implementations of Java ME and the Java ME TCK framework.

Back to top

Q: How are you making the Java HotSpot and compiler code available now?

A: The Java HotSpot and compiler code is available in zip files and also via a read-only Subversion repository. Access to the Subversion repository requires a java.net user ID, but getting one is free and easy. Visit https://www.dev.java.net/servlets/Join for more information.

Back to top

Q: What will happen to the JDK 6 and JDK 7 projects which are currently licensed under the Java Research License (JRL) on java.net?

A: Sun will continue the JDK6 and JDK7 projects under the JRL license at least until all source code is made available in early 2007. Afterwards we will determine if there is any interest in the community for continuing these projects—we expect there to be none. The J2SE 5.0 source code will not be released under an open-source license, and the "Tiger" Project will continue to be available, offering this code base under the JRL.

Back to top

Q: How do I make a suggestion or comment about Sun's open-source JDK effort?

A: Use the general feedback mailing list on openjdk.dev.java.net.

Back to top

Open-Source Communities and Java

Q: Have you been engaging with the non-Sun Java SE platform communities such as Apache Harmony, GNU Classpath, and Kaffe?

A: The Java developer ecosystem has a lot of very smart, experienced, community-savvy people who are passionate about the platform and eager to help. We've been in contact with these projects over the last few years, but in much more regular contact during the last few months. We've been talking with people like Geir Magnusson of Apache Harmony, Dalibor Topic of Kaffe, and Mark Wielaard of GNU/Classpath to test ideas and get their perspectives. We are very grateful to all of these people for the help and advice they have so generously and graciously offered.

Back to top

Q: Are you planning to work with these communities directly?

A: The Java ecosystem can support multiple implementations. Choice and differentiation keeps both commercial and open-source implementations on their toes, and we're not expecting any of the existing open-source Java SE or Java ME implementation communities to "close up shop" now that the JDK and Java ME implementations have been open sourced. It wouldn't be

good for Java technology if they did!

But we do hope that developers who are working in these and other open-source communities will consider joining the OpenJDK Community, the Mobile and Embedded Community, and the GlassFish Community (as, indeed, some Java EE implementors have already done). We're confident that developers will feel free to contribute to multiple projects, and through participation, allow good ideas to combine in new and interesting ways. Ultimately, we're hoping that the communities and implementations are differentiated by technology, not by license or ideology, for the betterment of the Java ecosystem as a whole.

At any rate, we hope to maintain a friendly, professional, and mutually beneficial relationship with all open-source Java SE platform development communities, and we look forward to finding ways to collaborate.

Back to top

Q: What impact will the choice of the GPL license have on your relationships with other open-source Java SE and Java ME implementation communities?

A: Because of the complexities of open-source licensing, not all open-source licenses are compatible. This means that unfortunately, once you choose a particular license, you close off collaboration that involves combining source code with communities using incompatible licenses. When we chose the GPL as the basis for Sun's open-source implementations, we made it easy to combine the open-source JDK code base with other GPL-licensed code bases such as GNU/Linux, GNU/Classpath, Kaffe, GNOME, and others. At the same time we made it hard to combine the code with projects licensed under the Apache Software License V2, such as the Apache Harmony project, because of license incompatibilities. By using the Classpath licensing exception we hope we have gone as far as we can to allow all Free and open source software to benefit from the OpenJDK Project code base.

We knew by selecting GPL that we would not please everyone. We picked this license because it tends to drive innovation into the open, and this open innovation will help to maintain and enhance the compatibility promise of Java technology. This license is also compatible with one of the largest and most influential open-source communities - namely, GNU/Linux - to broaden the Java developer community and enable it to enter new markets.

Back to top